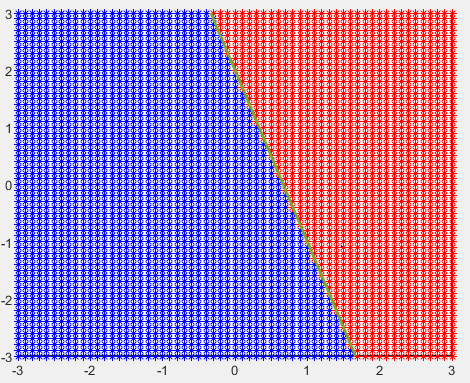
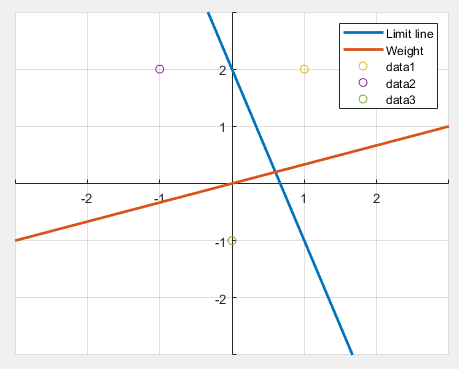
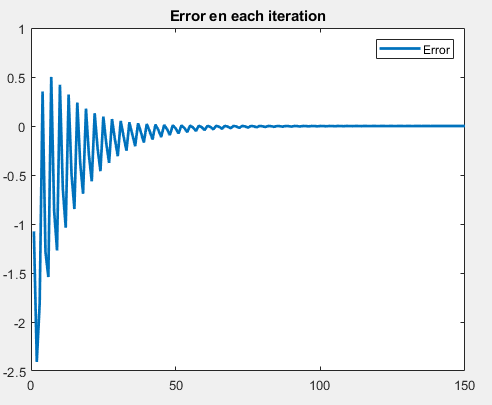
# Neural Networks Practice 4-Adaline

## Elizarraras Llanos Angel Gustavo

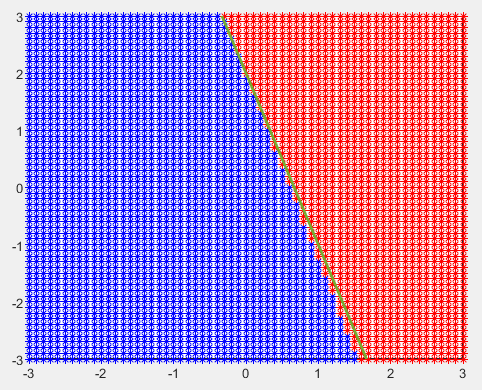
## Practice 1

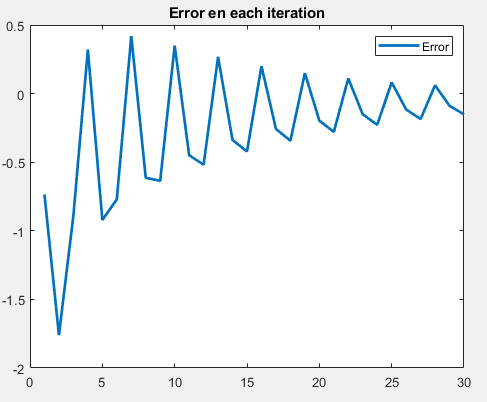
50 EPOCHS  






10 EPOCHS





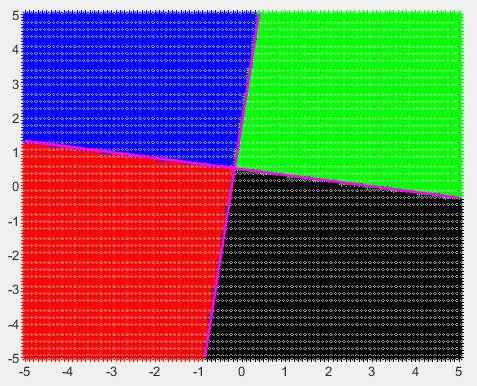
|  |
| --- |
| %% Training  for i = 1:Epochs  for j = 1:3    a = W\*P(:,j) + b;  e = Target(j) - a;  x = alpha\*e\*P(:,j)  W = W + x';  b = b + alpha\*e;  g(j) = e; % saves the value of error in a 1x3 vector  end  error = [error,g]; % concatenate g for the total error vector  end |

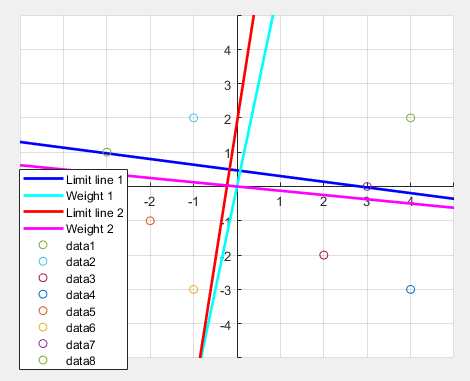
# Practice 2

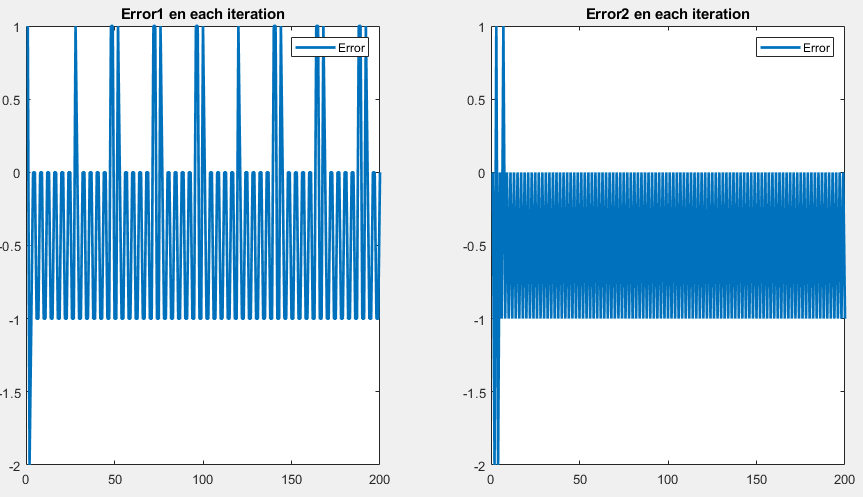
Designs and implement an artificial neural network able to separate four different classes.

Proposed classes:

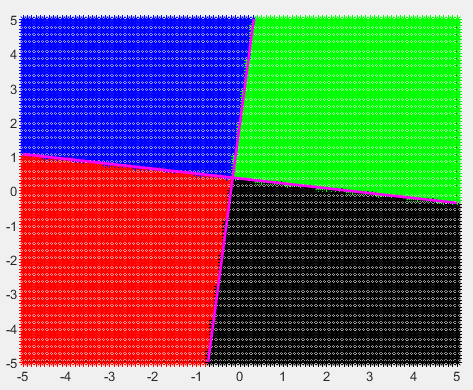
50 Epochs



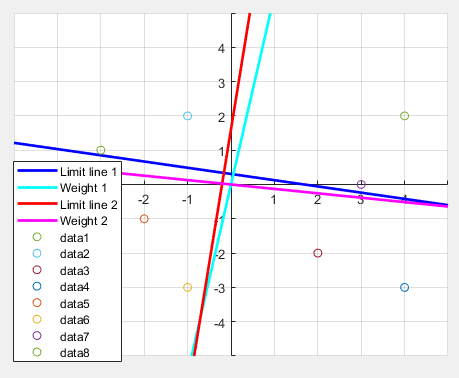


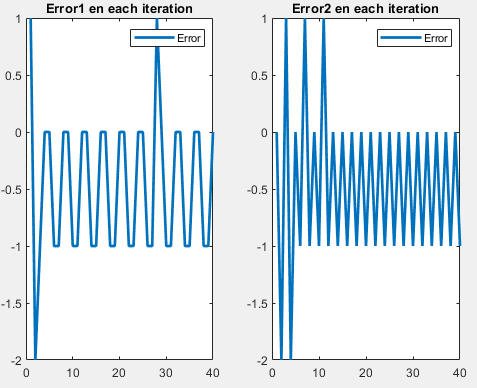


10 Epochs



I will add the other graph, because the lines are slightly moved.





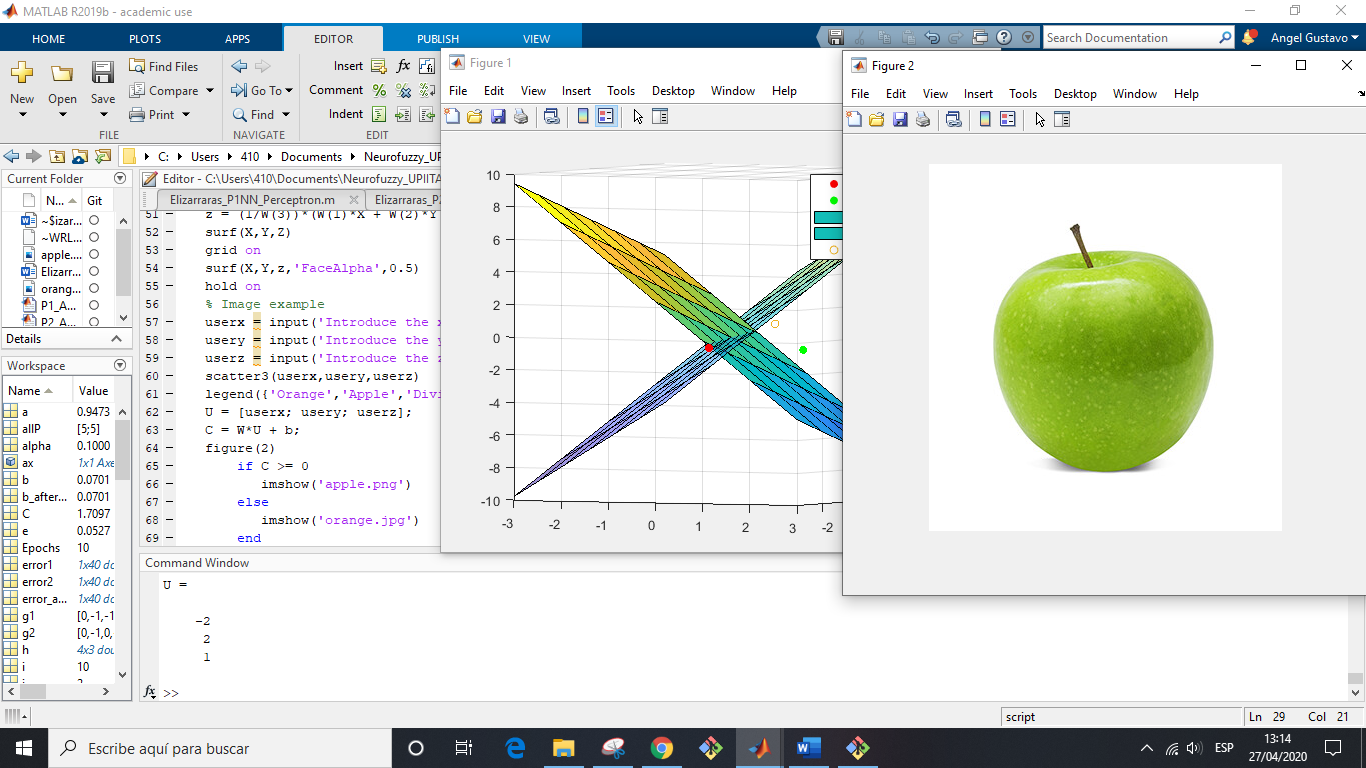
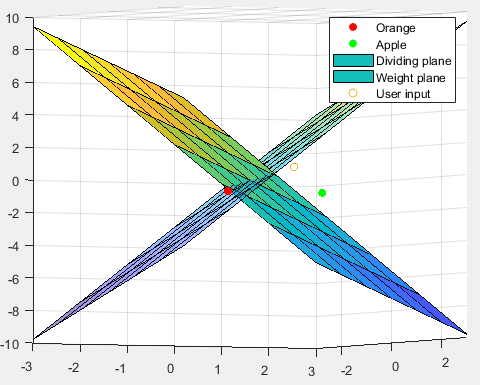
Error is a 2x1 matrix, so I plot both errors.

|  |
| --- |
| %% Training  for i = 1:Epochs  for j = 1:4    a(1) = dot(W(1,:),P(:,j)) + b(1);  a(2) = dot(W(2,:),P(:,j+4)) + b(2);  e = Targets(j,:) - hardlim(a);  x1 = alpha\*e(1)\*P(:,j);  x2 = alpha\*e(2)\*P(:,j+4);  W(1,:) = W(1,:) + x1';  W(2,:) = W(2,:) + x2';  b(1) = b(1) + alpha\*e(1);  b(2) = b(2) + alpha\*e(2);  g1(j) = e(1); %first 4 error 1 values  g2(j) = e(2); %first 4 error 2 values  end  error1 = [error1,g1];%fillling vector to plot  error2 = [error2,g2];%fillling vector to plot  end |

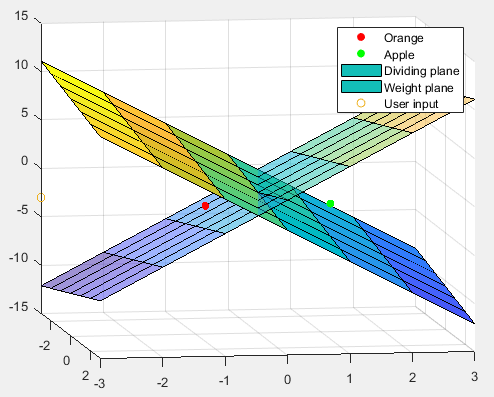
# Practice 3

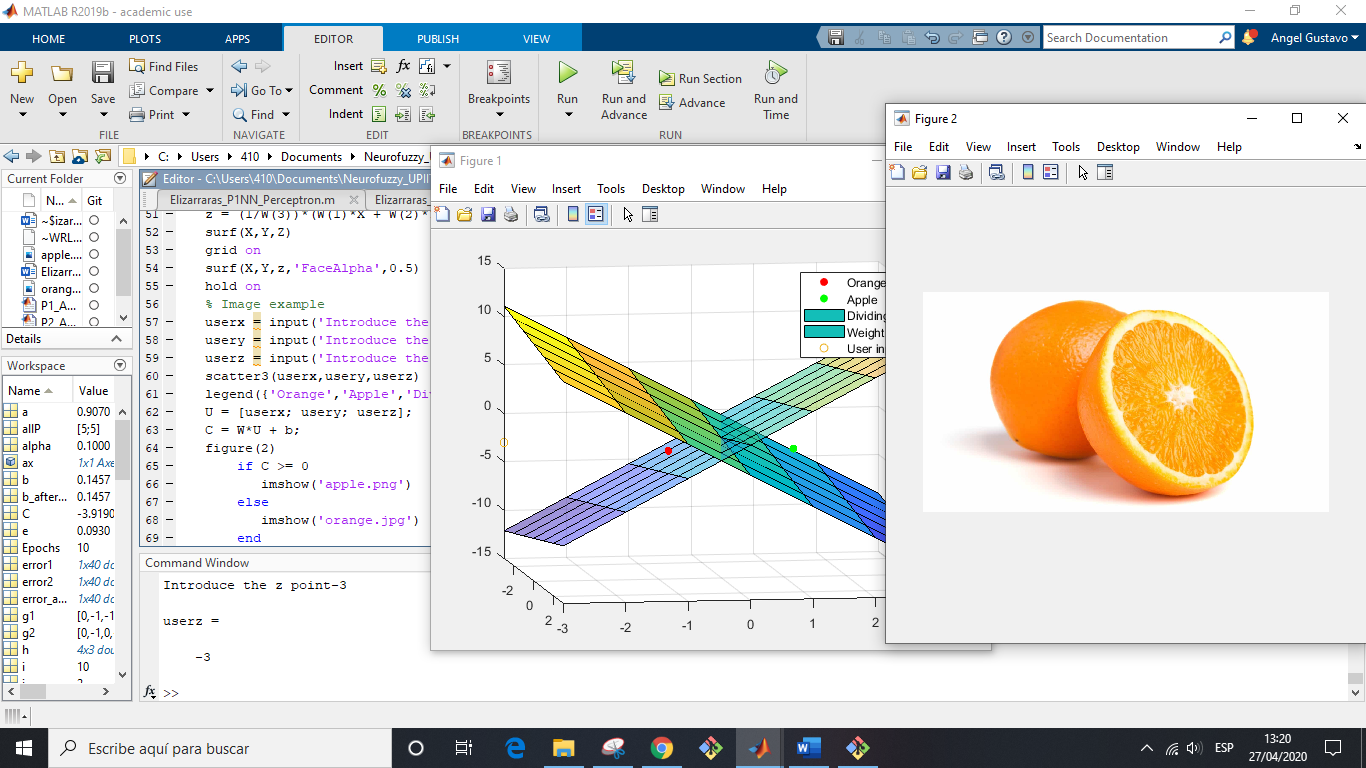
50 epochs

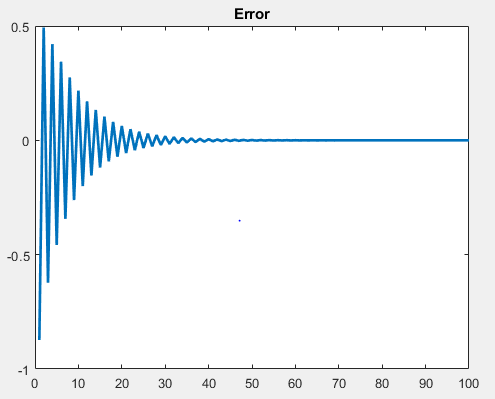
User input = [ -2 2 1 ], we notice is in the side of apple. The weight plane is the transparent one



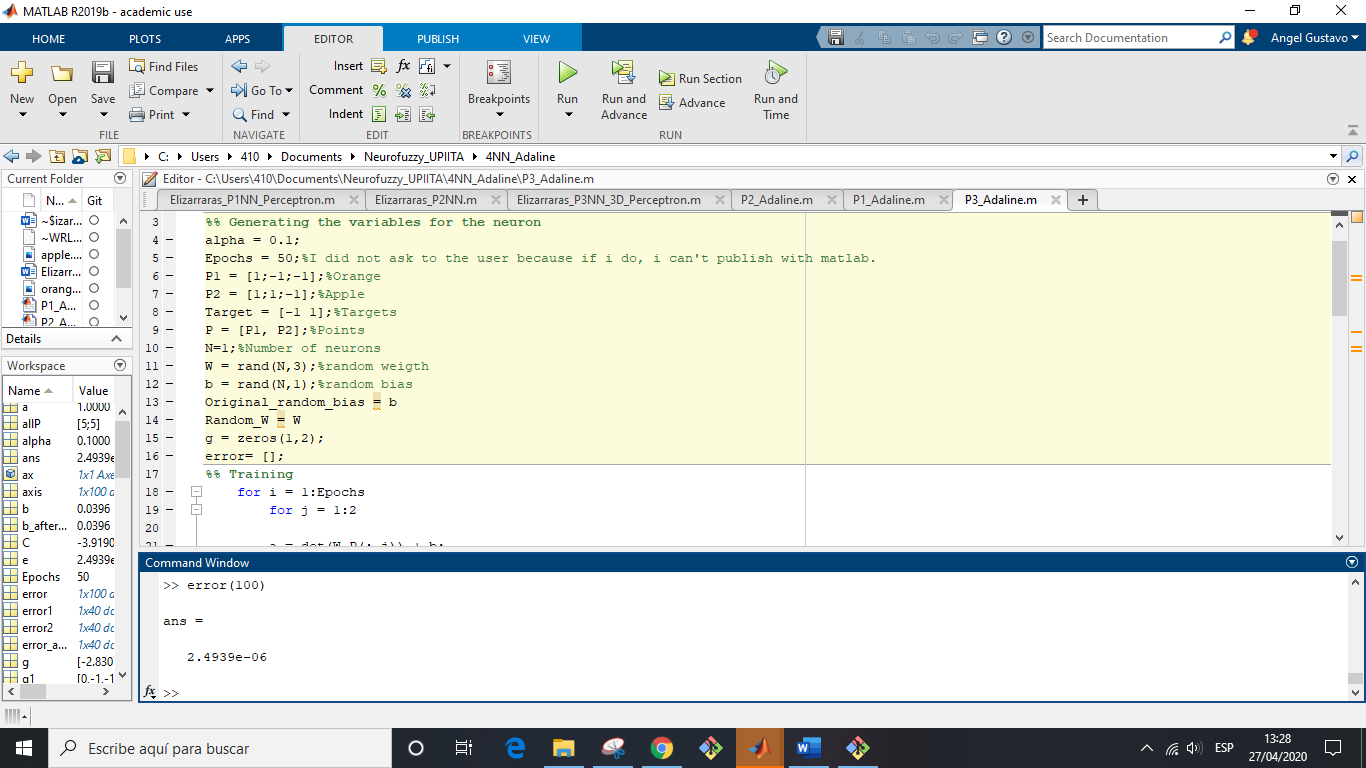
User input = [-3 -3 -3]. We notice that the input user is in the orange side.







The error is not cero



10 Epochs

